OPENCOCKPITS IOCards USBKeys

MANUAL DE INSTALACION Y USO

Opencockpits
All for your cockpit

INTRODUCCION

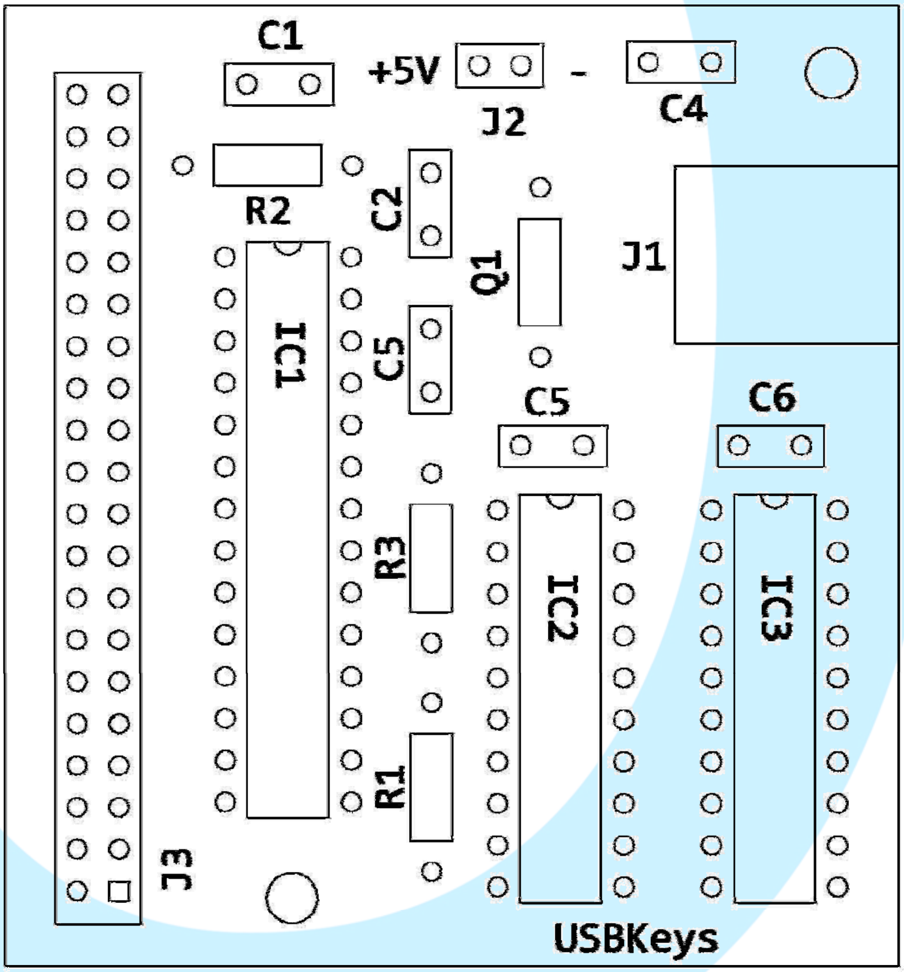
La USBKeys gestiona de forma automática teclados en matriz.

Esta placa permite gestionar y controlar un teclado mediante el conexionado de pulsadores en una matriz de 11x8 lineas, lo cual nos proporciona un total de 88 teclas.

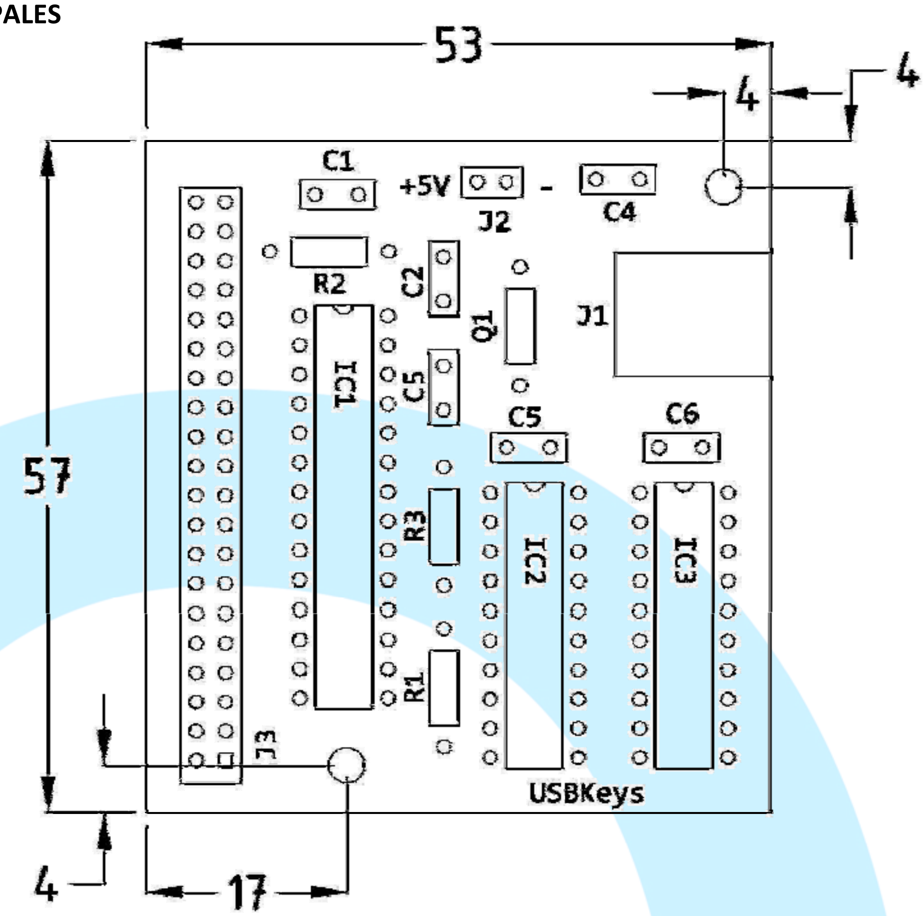
La placa se conecta al ordenador por puerto USB, lo cual nos permite una extrema facilidad a la hora de configurarla, asimismo se gestiona mediante el protocolo IOCP, con el lenguaje de programación SIOC o bien mediante un emulador de teclado por software.

LISTADO DE COMPONENTES

- C1, C4, C5, C6 = CONDENSADOR 0.1 uF
- C2, C3 = CONDENSADORES 22Pf
- IC1 = MICROCONTROLADOR 16C745
- IC2, IC3 = INTEGRADO 74HC541
- J1 = CONECTOR USB
- J2 = CONECTOR ALIMENTACION 2 PINES
- J3 = CONECTORES CABLE PLANO 40 PINES
- Q1 = CRISTAL DE CUARZO 6MHZ
- R2, R3 = RESISTENCIA 10K
- R1 = RESISTENCIA 1K5



DIMENSIONES PRINCIPALES



DESCRIPCION DE LOS CONECTORES:

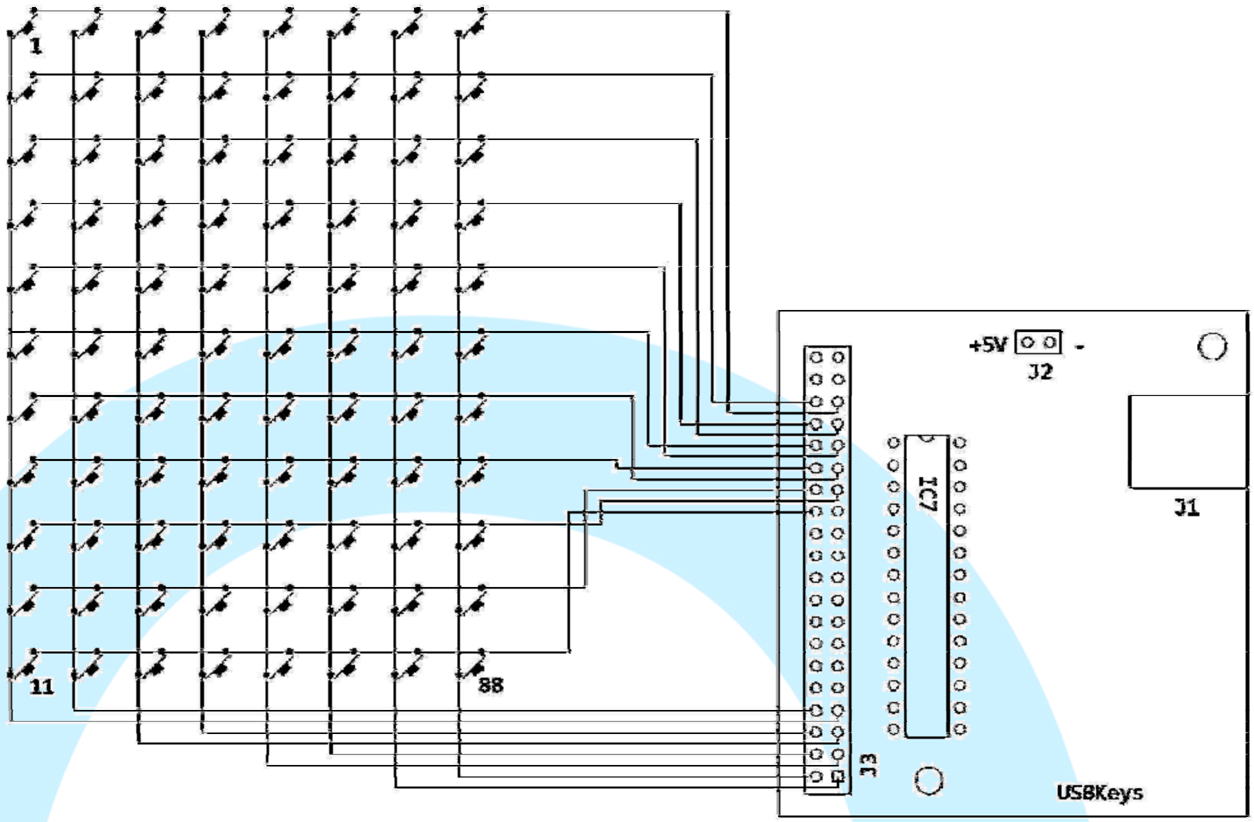
- J1 = Conector USB, permite la conexión al ordenador directamente, en el momento de conectarse el ordenador reconocerá la tarjeta y automáticamente instalará el driver para dispositivos HID.
- J2 = Conector de alimentación, no se utilizará.
- J3 = Conector 40 pines para el conexionado de la matriz (ver dibujo explicativo)

CONEXIONADO DE LOS PULSADORES:

Para realizar el cortocircuito de los diferentes pulsadores se hará en matriz, según la tabla siguiente:

Conector J3	7 (pin1)	8 (pin2)	5 (pin3)	6 (pin4)	3 (pin5)	4 (pin6)	1 (pin7)	2 (pin8)
K (pin26)	87	88	85	86	83	84	81	82
I (pin27)	71	72	69	70	67	68	65	66
J (pin28)	79	80	77	78	75	76	73	74
G (pin29)	55	56	53	54	51	52	49	50
H (pin30)	63	64	61	62	59	60	57	58
E (pin31)	39	40	37	38	35	36	33	34
F (pin32)	47	48	45	46	43	44	41	42
C (pin33)	23	24	21	22	19	20	17	18
D (pin34)	31	32	29	30	27	28	25	26
A (pin35)	7	8	5	6	3	4	1	2
B (pin36)	15	16	13	14	11	12	9	10

Para una mejor explicación del conexionado, a continuación se adjunta una imagen:



En el caso de cortocircuitar varios pulsadores de la matriz a la vez, la placa gestionará las primeras 5 teclas pulsadas, y serán lanzadas de forma secuencial, perdiéndose a partir de ahí el resto de teclas.

ATENCION: NUESTRO CONEXIONADO NO DEBE NUNCA CORTOCIRCUITAR LINEAS DEL MISMO GRUPO

SOFTWARE

En el caso de que queramos conectar la placa mediante un emulador de teclado por software usaremos el encoder_keys.exe

Asimismo podríamos usar directamente SIOC para gestionar nuestro teclado, ya que desde la versión 3.52b de SIOC ya es posible enlazar la tarjeta y emular un teclado desde el, la opción es la mas aconsejable, por la versatilidad del mismo lenguaje.

ENCODER_KEYS: EMULADOR DE TECLADO

Mediante el fichero encoder_keys.ini se configura el acceso a la tarjeta.

Parámetros en dicho fichero:

MUSB=No

Pondremos a Yes en el caso de que tengamos conectadas mas tarjetas en el mismo ordenador.

deviceUSB=2048

En este caso le diremos al controlador en que numero de dispositivo debe conectarse.

Para conocer el numero de Device, conectaremos la placa anotaremos el numero que nos indica el propio controlador. En cada puerto USB el numero es diferente.

Window = “a.txt – Bloc de notas”

Pondremos en esta línea la ventana a donde queramos que se dirijan las pulsaciones de teclas. Para ello la ventana deberá estar en primer plano, en ningún caso minimizada o ejecutandose tras otra ventana.

[Asignacion de teclas]

#1=A
#2=B
#3=C
…

En esta forma enviaremos los códigos de las teclas a las secuencias de teclas.

En el ejemplo de arriba, se ve claramente que al pulsar la tecla 1 (#1, conectada en la matriz como pin7+pin35) se enviara a la ventana asignada la letra “A”, al pulsar la 2 (#2, conectada en la matriz como pin8+pin35) se enviara la letra “B”, etc…

Teclas combinadas como Ctrl, Alt, etc… se envían de una forma predeterminada, como veremos al final del documento.

Cuando ejecutamos el programa automáticamente se minimizara en la barra de estado (tray), con el icono que parece y el botón derecho podemos maximizarlo otra vez, en la ventana asimismo nos aparece la ultima tecla pulsada y el numero de Device.

USO DE SIOC

Para usar nuestra tarjeta USBKeys con SIOC, deberemos “declarar” dicha tarjeta en nuestro fichero sioc.ini, para ello añadiremos o editaremos la línea siguiente:

USBKeys=XX,YY

Donde XX es el numero de identificador de la tarjeta e YY es el numero de Device propio del USB.

Como ejemplo, si conectamos una tarjeta USBKeys en el Device 34, entonces la declararemos como sigue:

USBKeys=1,34

(usamos el numero 1 para el identificador, pero podríamos usar cualquier otro, la única condición es que no se repita dentro del mismo sioc.ini)

Las combinaciones de teclas y sus códigos se parametrizan igual que si usásemos el encoder_keys.exe

A continuación un pequeño ejemplo para enviar mediante SIOC, las pulsaciones Ctrl+Alt+5 y Ctrl+Alt+6, mediante un encoder.

Para ello asignaremos en el sioc.ini la combinación de teclas a los códigos 70 y 71 bajo el parámetro [Asignacion de teclas]:

#70=\3\15\2\4
#71=\3\16\2\4

Como podemos ver (al final del documento están las asignaciones especiales), en el código 70 hemos aplicado la secuencia Ctrl pulsado (\3) Alt pulsado (\1) tecla 5 (5) Alt liberado (\2) y Ctrl liberado (\4) y en el código 71 lo mismo pero con la tecla 6 (6), esto significa que cuando el encoder gire hacia un lado nos mandara a la ventana activa la secuencia Ctrl+Alt+5 y hacia el lado contrario nos mandara la secuencia Ctrl+Alt+6

```
// ******************************************************************************
// * Config_SIOC ver 3.6B2    -      By Manolo Vélez    -    www.opencockpits.com
// ******************************************************************************
// * FileName : teclas ctrl alt 5 y 6.txt
// * Date : 16/02/2009

Var 0001, Link IOCARD_ENCODER, Input 10, Aceleration 1, Type 2
{
  IF V0001 = 1
  {
    V0002 = 70
    V0002 = 0
  }
  ELSE
  {
    V0002 = 71
    V0002 = 0
  }
}

Var 0002, Link KEYS
```

Asignamos un valor 0 a la pulsación después de cada asignación de código, para evitar que al estar el encoder conectado nos este repitiendo ese envio, asi con el 0 se detiene el envio de datos a la tarjeta.

Si ejecutamos ahora SIOC, con la tarjeta correctamente configurada y las conexiones realizadas, entonces veremos que al girar el encoder nos estará mandando esas pulsaciones a la ventana adecuada, y esto nos sirve perfectamente para casos en los que no conocemos los offsets adecuados y solo tenemos conocimiento de las pulsaciones de teclado con que funciona la utilidad (en el caso de PMDG, y otros como Wilco, etc… nos es muy importante)

DEFINICION DE TECLAS ESPECIALES

“\” +	A	B	C	D	E	F	G	H	I	J	K	L	M
Funcion	BKSP	TAB	ENTER	ESC	F1	F2	F3	F4	F5	F6	F7	F8	F9

“\” +	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Funcion	F10	F11	F12	HOME	END	UP	DOWN	LEFT	RIGHT	PG UP	PG DN	INS	DEL

“\” +	1	2	3	4	5	6	\
Funcion	SHIFT DN	SHIFT UP	CTRL DN	CTRL UP	ALT DN	ALT UP	\

DEFINICION DE TECLAS ESPECIALES EN BLOQUE NUMERICO

“/” +	/	0	1	2	3	4	5	6	7	8	9	A
Funcion	/	NUM 0	NUM 1	NUM 2	NUM 3	NUM 4	NUM 5	NUM 6	NUM 7	NUM 8	NUM 9	MULTIPLY

“/” +	B	C	D	E	F	G
Funcion	ADD	SEPARATOR	SUBSTRACT	DECIMAL	DIVIDE	SPACE

Fíjese que el símbolo que anteponemos a la función del bloque numérico no es el mismo que para las funciones especiales, en el primer caso es la barra invertida “\” y para el Segundo es la barra normal “/”.

Si anteponemos el símbolo “<”a cualquier letra, conseguiremos enviarla como minúscula, porque a veces según que software distingue entre mayúsculas y minúsculas.

Ejemplos:

Ctrl+Shift+F8 = \3\1\L\2\4

Ctrl+Shift+L = \3\1L\2\4

Ctrl+numpad 5 = \3/5\4

Minúscula “a” = <A

Dese cuenta que en el primer caso la letra “L” va acompañada de “\” lo cual significa que el código que enviamos es F8, y en el Segundo caso la “L” no va acompañada de ningún carácter, por lo tanto simplemente enviamos la letra “L”.

En el tercer caso la barra normal nos envía a la definición de teclas del bloque numérico.

Y, por último, en el cuarto caso, enviamos una “a” minúscula, ya que le hemos antepuesto el símbolo “<”, porque por defecto las letras se envían en mayúscula.